

TDD Eclipse

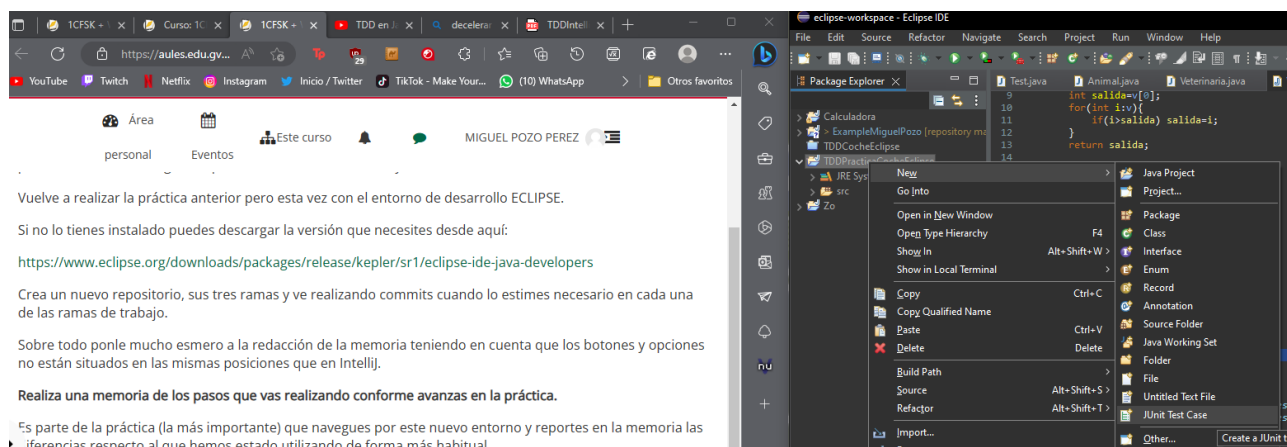
TDD cycle



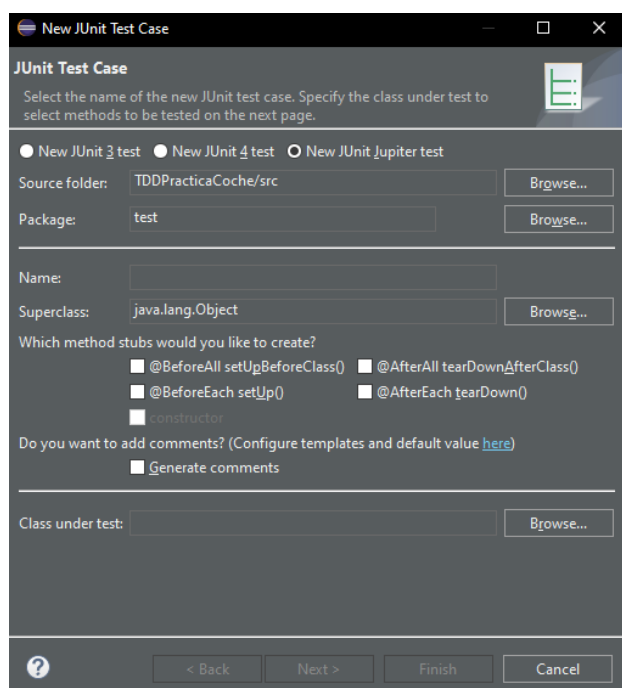
Curso	1W DAW
Alumno	Miguel Pozo Pérez
Profesor/a	Jaime García Quiles

Practica

En eclipse es algo diferente ya que nos sale directamente la opción de crear el test.



Elegimos el Junit deseado y le damos a finalizar



Creamos los test para probar Junit y en los errores que nos aparecen indicamos que nos hagan tanto la variable velocidad como los métodos acelerar y desacelerar como ya hemos visto en IntelliJ

The screenshot shows a web browser on the left with a page titled "Entrega de la práctica:" and an Eclipse IDE on the right showing Java code for a car simulation.

Entrega de la práctica:

- Copia un enlace al repositorio GitHub donde has subido la práctica.
- Asegúrate de que cada rama contiene lo que se pide.
- Crea una rama que se llame "Memoria" y sube a esta rama la memoria realizada. *** FORMATO PDF O WORD ***

Tiempo estimado para realizar la práctica: 2h:15'. Los 30 minutos extra son por si tienes que instalar ECLIPSE y puesto que hay que bichear un poquito por los menús.

Esta URL es un video de hace unos años, pero es válido a nivel general, de un pequeño ejemplo de TDD con CLIPSE.

Te recomiendo NO importar el proyecto, si no volver a teclear todo el código, pero tú mism@.

Estado de la entrega

Estado de la entrega	No entregado

The Eclipse IDE shows the following code in `Coche.java`:

```

package test;

import static org.junit.jupiter.api.Assertions.fail;

import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;

class CocheTest {

    @Test
    public void testCrearCocheVelocidadCeroMiguelPozo() {
        Coche nuevoCoche = new Coche();
        Assertions.assertEquals(0, nuevoCoche.velocidad);
    }

    @Test
    public void testAcelerarCocheVelocidadIncrementaMiguelPozo() {
        Coche nuevoCoche = new Coche();
        nuevoCoche.acelerar(10);
        Assertions.assertEquals(10, nuevoCoche.velocidad);
    }

    @Test
    public void testDecelerarCocheVelocidadDisminuyeMiguelPozo() {
        Coche nuevoCoche = new Coche();
        nuevoCoche.velocidad = 10;
        nuevoCoche.decelerar(10);
        Assertions.assertEquals(0, nuevoCoche.velocidad);
    }

    @Test
    public void testAlDecelerarCocheVelocidadNoPuedeSerMenorQueCeroMiguelPozo() {
        Coche nuevoCoche = new Coche();
        nuevoCoche.velocidad = 10;
        nuevoCoche.decelerar(10);
        Assertions.assertEquals(0, nuevoCoche.velocidad);
    }
}

```

Por ultimo actualizamos el método decelerar para que no salgan valores negativos y se valide el test.

The screenshot shows the same web browser on the left and an updated Eclipse IDE on the right.

Entrega de la práctica:

- Copia un enlace al repositorio GitHub donde has subido la práctica.
- Asegúrate de que cada rama contiene lo que se pide.
- Crea una rama que se llame "Memoria" y sube a esta rama la memoria realizada. *** FORMATO PDF O WORD ***

Tiempo estimado para realizar la práctica: 2h:15'. Los 30 minutos extra son por si tienes que instalar ECLIPSE y puesto que hay que bichear un poquito por los menús.

Esta URL es un video de hace unos años, pero es válido a nivel general, de un pequeño ejemplo de TDD con CLIPSE.

Te recomiendo NO importar el proyecto, si no volver a teclear todo el código, pero tú mism@.

Estado de la entrega

Estado de la entrega	No entregado

The Eclipse IDE shows the updated code in `Coche.java`:

```

package test;

public class Coche {
    public Object velocidad;

    public void acelerar(int aceleracion) {
        // 1000 Auto-generated method stub
        velocidad += aceleracion;
    }

    public void decelerar(int deceleracion) {
        // 1000 Auto-generated method stub
        velocidad -= deceleracion;
        if (velocidad < 0) velocidad = 0;
    }
}

```